



OpenGL ES 3.0 – Challenges and Opportunities

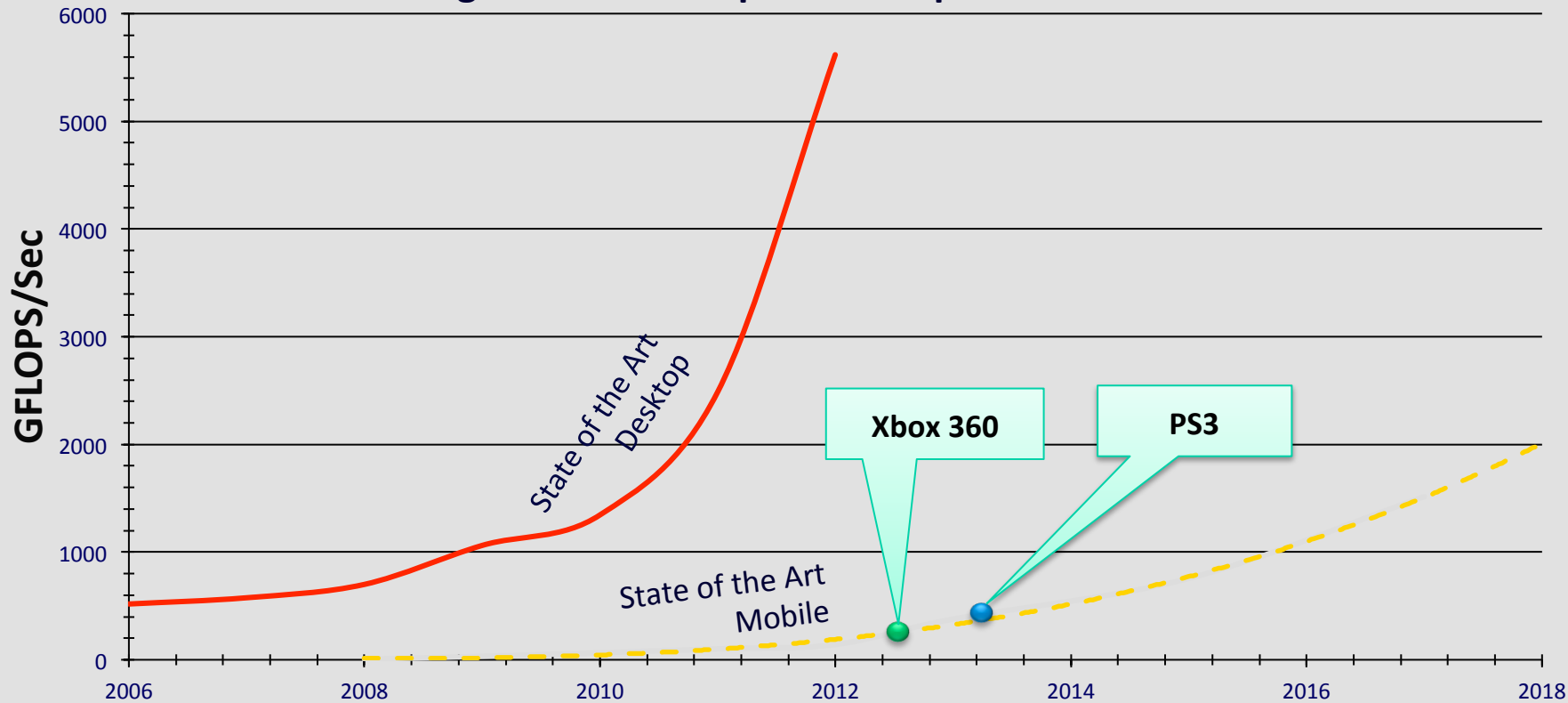
Marc Hehmeyer, CTO

August, 2013

“It’s unquestionable that within a very short time, we’re going to have portable cell phones that are more powerful than the current-gen consoles”

John Carmack 07/2011

How long before Desktop GPU compute is in Mobile?



Why do we care?



NFS Underground 2, EA Black Box



Half Life 2, Valve Corporation



GTA 3 San Andreas, Rockstar North



DOOM 3, Id Software

We've come a long way!



OpenGL ES

- **Open Graphics Library for Embedded Systems**
- **Low-Level software Interface to graphics hardware**
- **Subset of OpenGL**

- **Various use of OpenGL ES powered GPUs**
 - Smartphones / tablets
 - TVs
 - Automotive
 - Many more



History of OpenGL ES

2003: OpenGL ES 1.0



Galaxy on Fire 2 SD, Fishlabs

History of OpenGL ES

2007: OpenGL ES 2.0



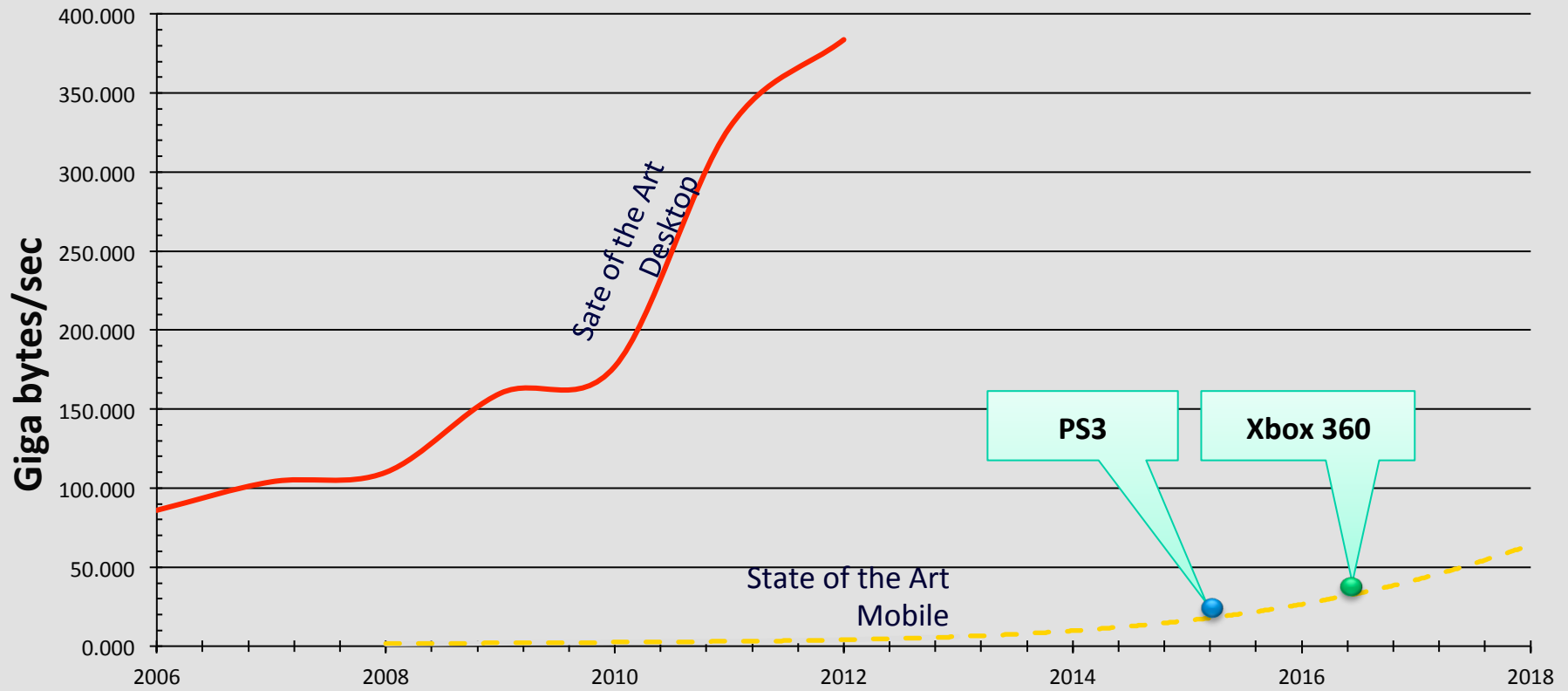
Galaxy on Fire 2 HD, Fishlabs

History of OpenGL ES



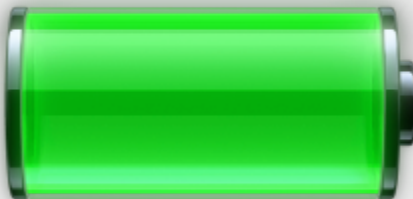
Galaxy on Fire, Fishlabs

How long before Desktop GPU Bandwidth is seen in Mobile?



OpenGL ES 3.0

- 2012: Specification released
- Feature set based on OpenGL 3.3 / 4.x
- Reduces need for extensions
- Full backward compatible with OpenGL ES 2.0



OpenGL ES 3.0

Transform
Feedback Mode

Vertex Array Objects

Pixel Buffer Objects

GLSL ES 3.00

**ETC2 Texture
Compression**

**Boolean Occlusion
Queries**

**Instanced
Rendering**

**Multiple Render
Targets**

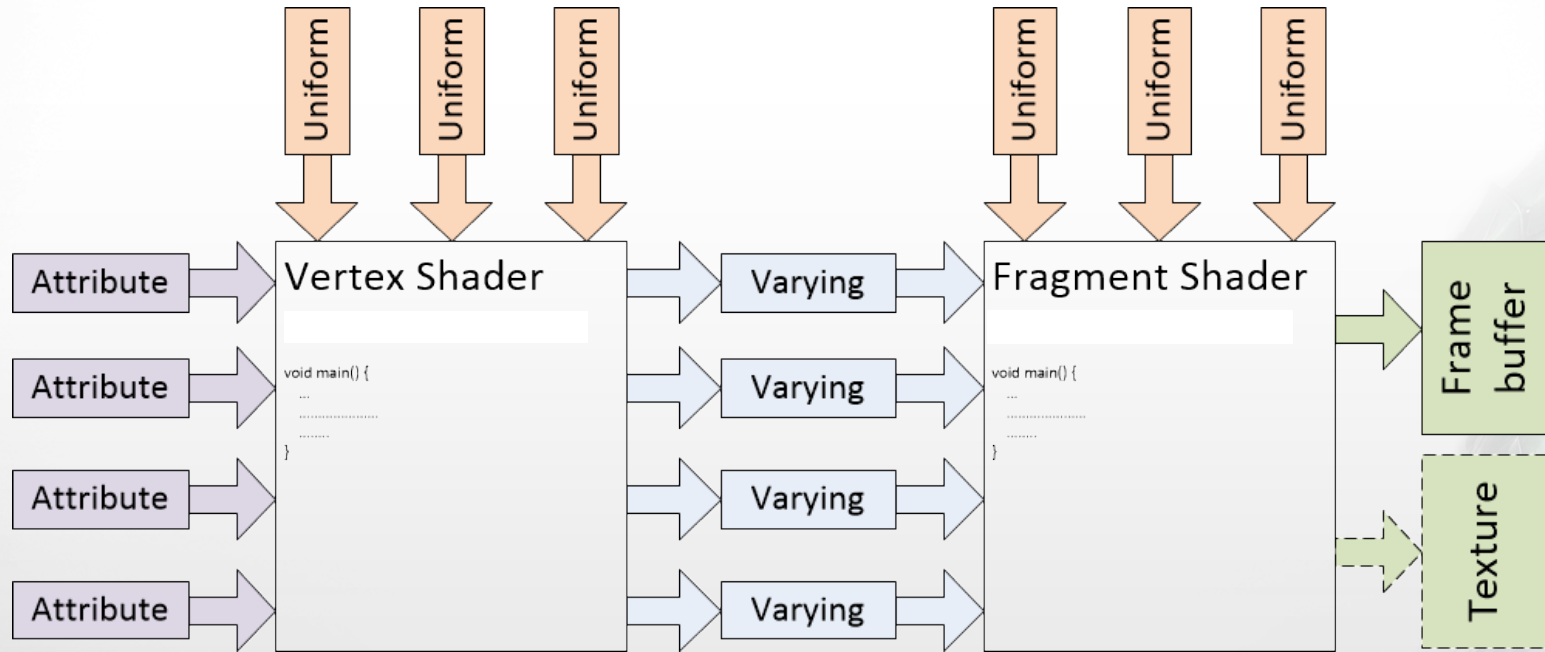
Uniform Buffer
Object

Sampler Objects

Sync Objects

And many
more ...

Shading Language GLSL ES 3.00



Shading Language GLSL ES 3.00

// Vertex shader

#version 100

```
uniform mat4 u_matViewProjection;  
attribute vec4 a_position;  
attribute vec2 a_texCoord0;  
varying vec2 v_texCoord;
```

```
void main(void) {  
    gl_Position = u_matViewProjection * a_position;  
    v_texCoord = a_texCoord0;  
}
```

// Fragment shader

#version 100

```
varying vec2 v_texCoord;  
uniform sampler2D s_baseTexture;
```

```
void main() {  
    gl_FragColor = texture2D(s_baseTexture, v_texCoord);  
}
```

// Vertex shader

#version 300 es

```
uniform mat4 u_matViewProjection;  
in vec4 a_position;  
in vec2 a_texCoord0;  
out vec2 v_texCoord;
```

```
void main(void) {  
    gl_Position = u_matViewProjection * a_position;  
    v_texCoord = a_texCoord0;  
}
```

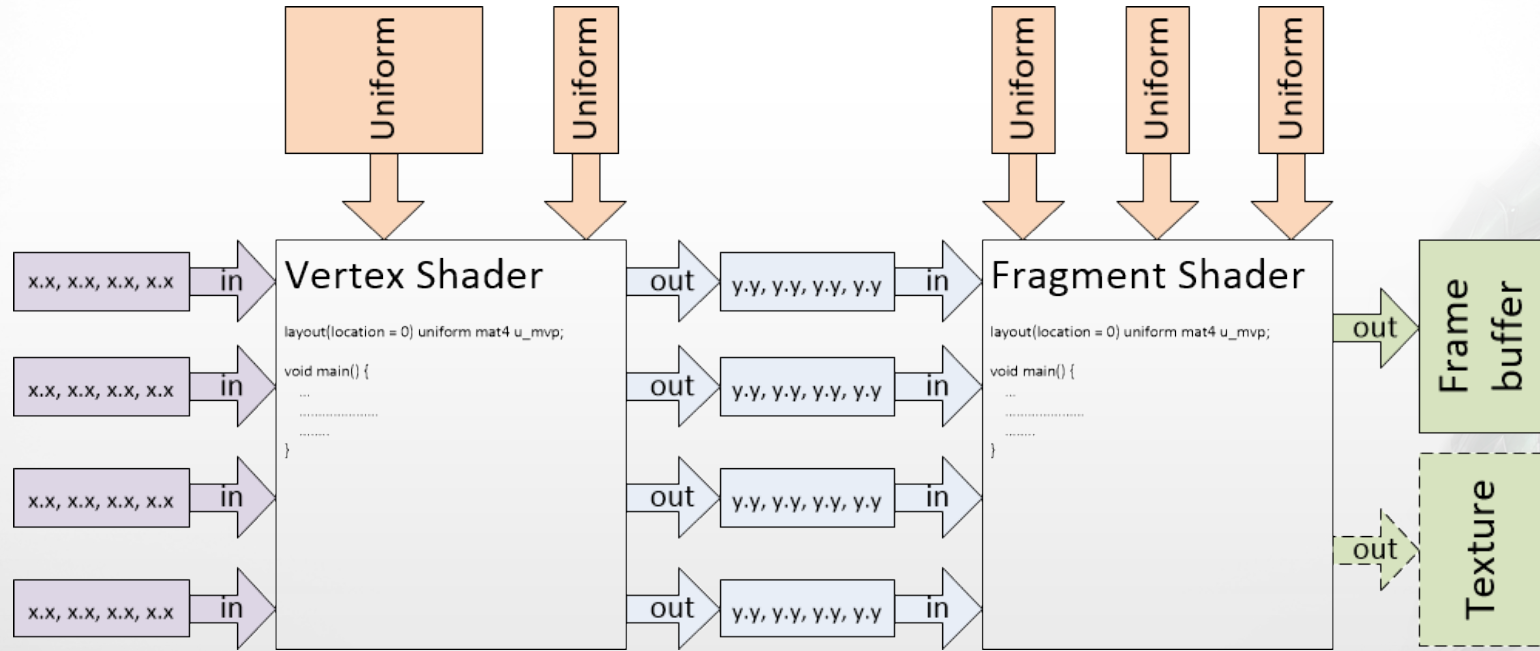
// Fragment shader

#version 300 es

```
in vec2 v_texCoord;  
uniform sampler2D s_baseTexture;  
layout(location = 0) out lowp vec4 outColor;
```

```
void main() {  
    outColor = texture2D(s_baseTexture, v_texCoord);  
}
```

Shading Language GLSL ES 3.00



ETC2 Texture Compression

- **Standard texture compression**
- **Support of Alpha, one and two channels**
- **Eliminate the limitations of ETC1**
 - No Alpha support
 - Poor texture quality
- **Theoretically no more need for proprietary texture formats**
 - Smaller filesize
 - No different asset packages

No graphical intense game without texture compression

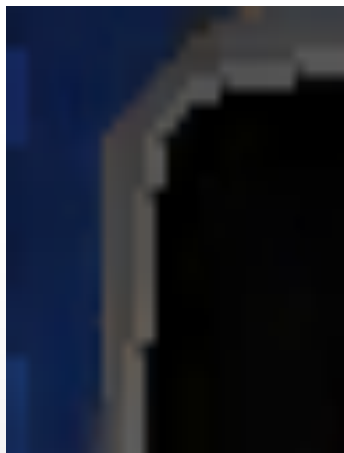
ETC2 Texture Compression



RGBA
32 BPP



PVRTC
2 BPP



PVRTC
4 BPP



ETC1
4 BPP



ETC2
4 BPP

Boolean Occlusion Queries

Software interface for hardware based visibility test

glGenQueries

glDeleteQueries

glBeginQuery

glEndQuery

glGetQueryObjectuiv



Boolean Occlusion Queries

```
...
int qid[NUM_OBJECTS];
unsigned int result = 0;

glGenQueries(NUM_OBJECTS, &qid[0]);

for (int i = 0; i < NUM_OBJECTS; ++i) {

    glBeginQuery(GL_ANY_SAMPLES_PASSED, qid[i]);

    // render objects with low details

    glEndQuery(GL_ANY_SAMPLES_PASSED);

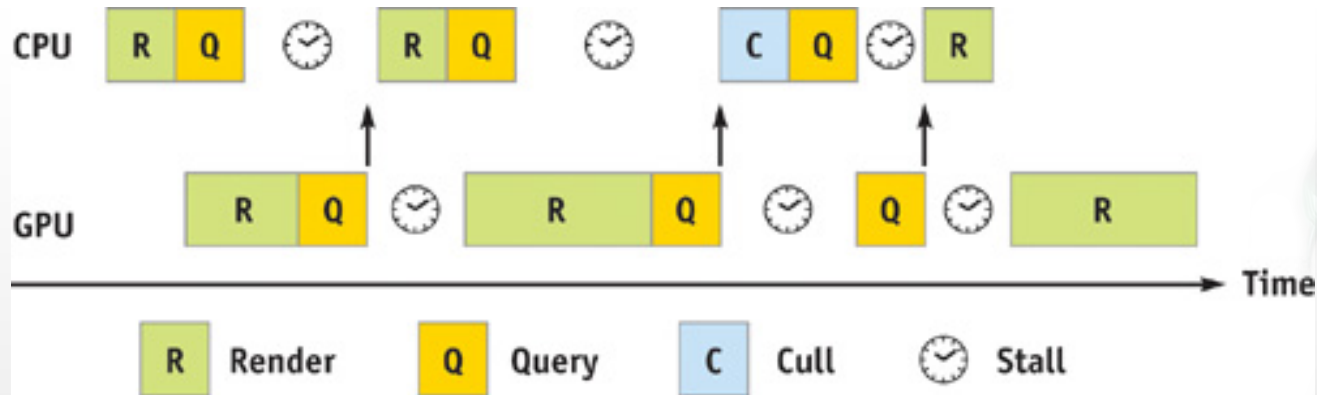
    while (result == GL_FALSE) {
        glGetObjectuiv(qid[i], GL_QUERY_RESULT_AVAILABLE, &result);
    }

    glGetObjectuiv(qid[i], GL_QUERY_RESULT, &result);

    if (result == GL_TRUE) {
        // render Objekt with full details
    }
}
...
```

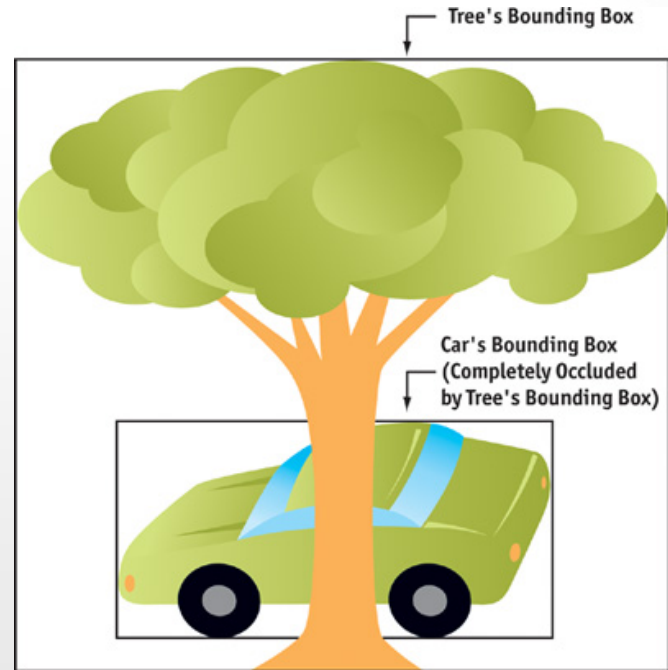
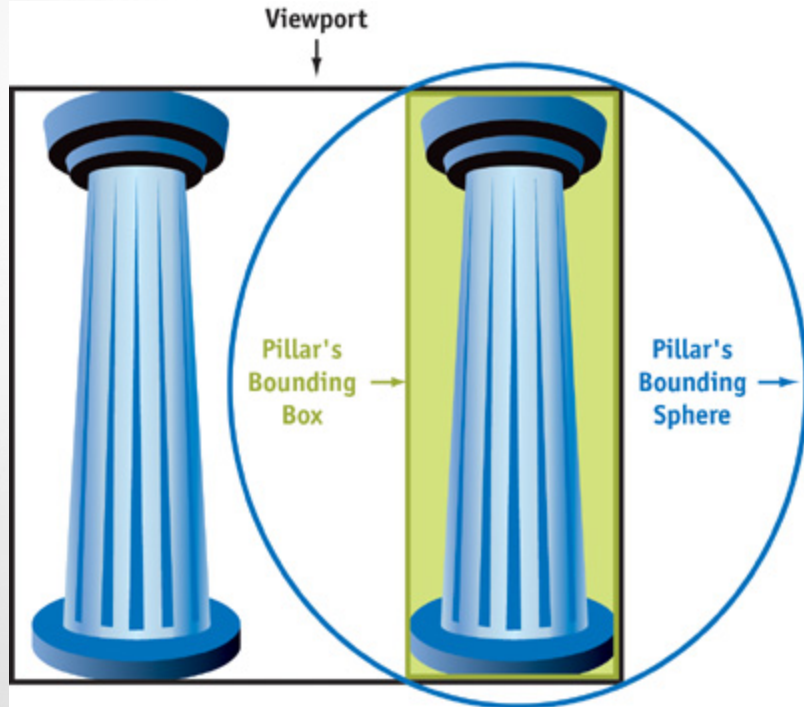
← Bad Idea

Boolean Occlusion Queries

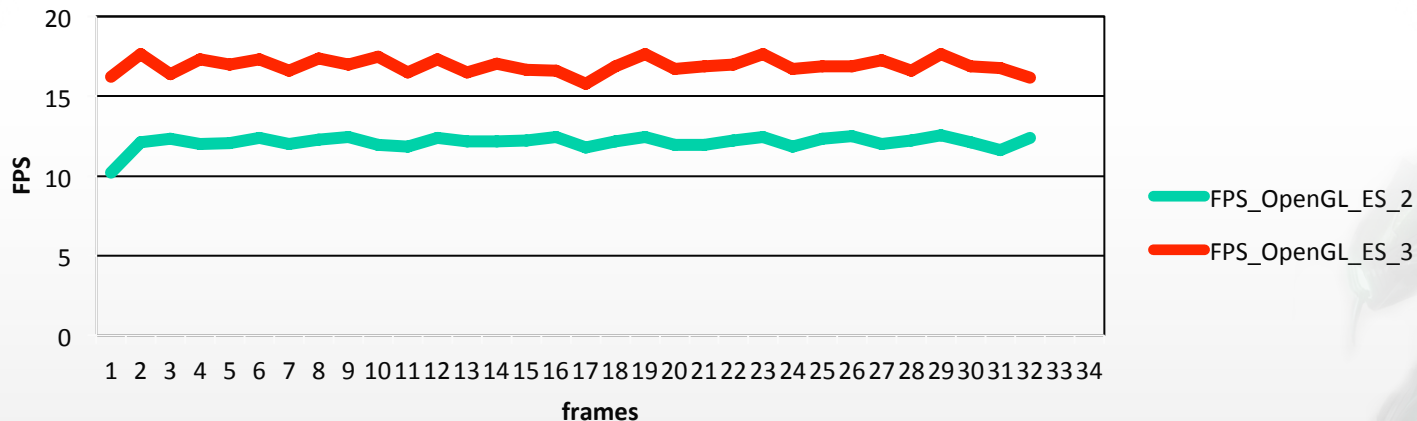


http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter06.html

Boolean Occlusion Queries

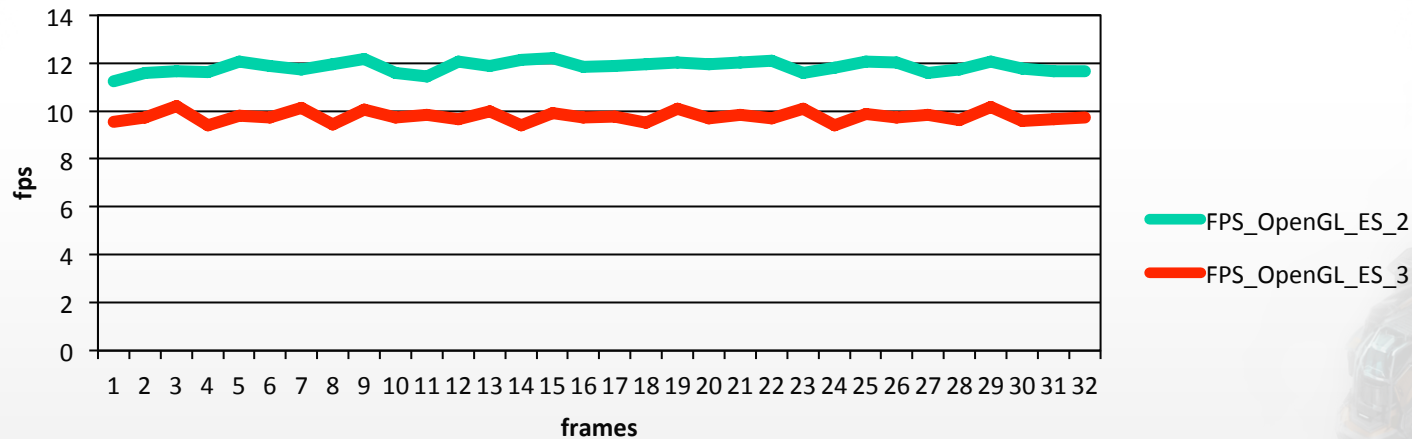


Boolean Occlusion Queries



90% of hidden geometry

Boolean Occlusion Queries



40% of hidden geometry

Boolean Occlusion Queries



Minecraft by Mojang

Instanced Rendering

- Minimize draw calls
- Powerful for scenes with a lot of identical geometries
- Lean interface

glDrawArraysInstanced(GLenum mode, GLint first, GLsizei count, GLsizei primcount)

glDrawElementsInstanced(GLenum mode, GLsizei count, GLenum type, const void indices, GLsizei primcount)*

glVertexAttribDivisor(GLuint index, GLuint divisor)

gl_InstanceID

- Might be hard to implement into existing rendering pipeline

Instanced Rendering

OpenGL ES 2.0

```
for ( int i = 0; i < numInstances; i++ ) {  
    // set for each instance the model-view-projection matrix  
    glDrawElements(GL_TRIANGLES, mesh->indx_count, GL_UNSIGNED_SHORT, mesh->indx);  
}
```

OpenGL ES 3.0

```
glDrawElementsInstanced(GL_TRIANGLES, mesh->indx_count, GL_UNSIGNED_SHORT, mesh->indx, numInstances);
```

Instanced Rendering

```
// Vertex shader
#version 100
uniform mat4 u_matViewProjection;
attribute vec4 a_position;
attribute vec2 a_texCoord0;
varying vec2 v_texCoord;
```

```
MVP = glGetUniformLocation( programObj, "u_matViewProjection" );

glUniformMatrix4fv(MVP, 1, GL_FALSE, &mvMatrix );
```

```
// Vertex shader
#version 100
attribute mat4 u_matViewProjection;
attribute vec4 a_position;
attribute vec2 a_texCoord0;
varying vec2 v_texCoord;
```

```
MVP = glGetAttribLocation ( programObj, "u_matViewProjection" );

for (int i = 0; i < 4; i++) {

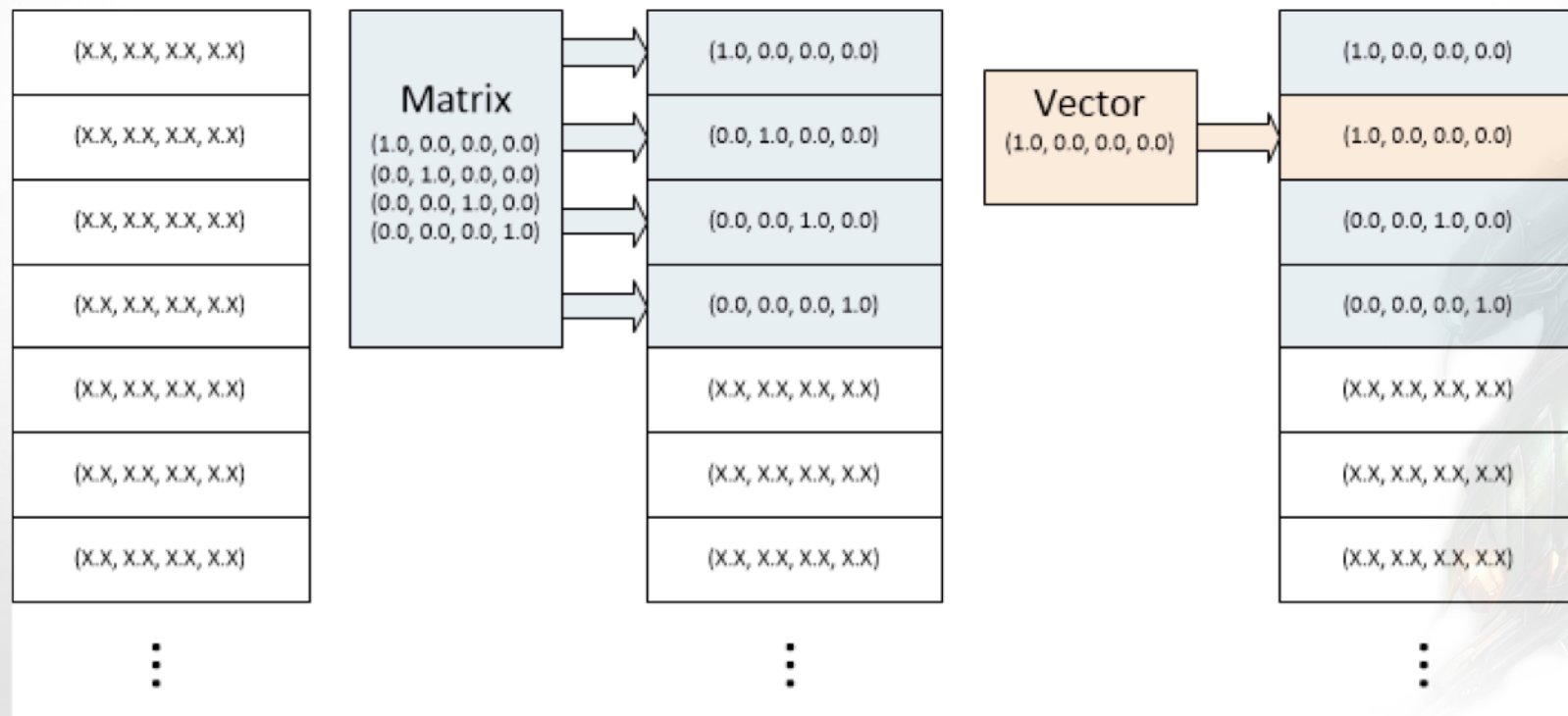
    glEnableVertexAttribArray(MVP + i);

    glVertexAttribPointer(MVP + i,
        4, GL_FLOAT, GL_FALSE,
        16*sizeof(GLfloat),
        &(matArray + 4*i*sizeof(GLfloat)));

    glVertexAttribDivisor(MVP + i, 1);

}
```

Instanced Rendering



Instanced Rendering

```
#define LTP_ARRAY      0
#define VERTEX_ARRAY  4

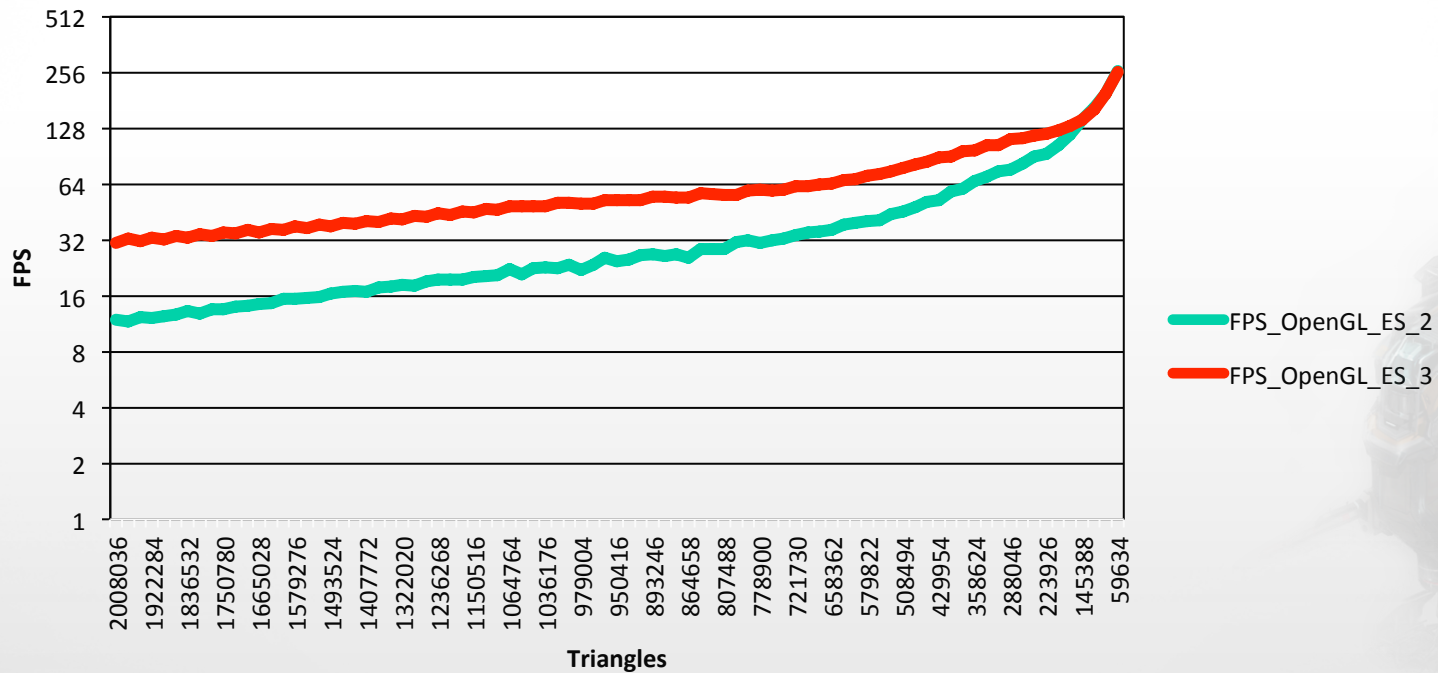
layout(location = LTP_ARRAY)    in highp mat4 inLocalToProjection;
layout(location = VERTEX_ARRAY) in highp vec3 inVertex;

void main() {
    gl_Position = inLocalToProjection * inVertex;
}
```

Instanced Rendering



Instanced Rendering

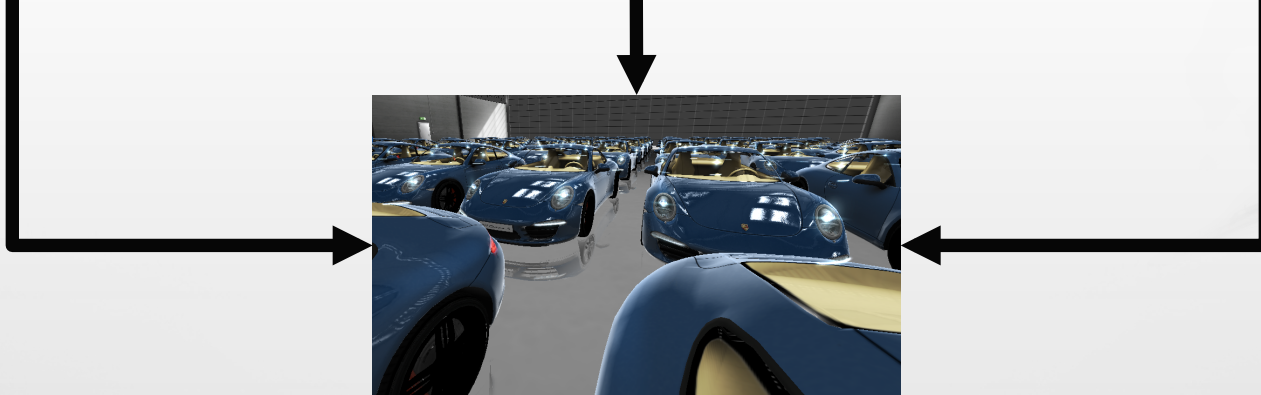
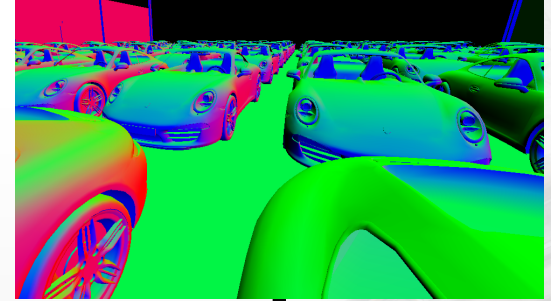
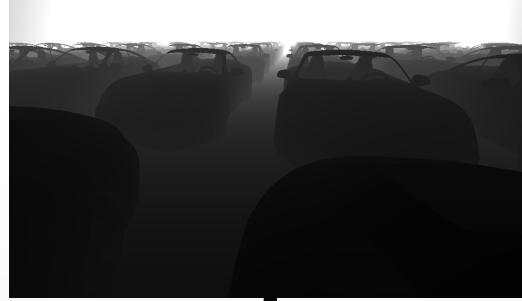


Multiple Render Targets (MRT)

- Render to multiple buffers in a single draw call
- Offers the possibility to perform next-gen visual effects in real-time
 - Deferred Lighting
 - Cel Shading
 - Deferred Decals
 - Real-time Local Reflections
 - Many more



Multiple Render Targets (MRT)



Multiple Render Targets (MRT)

```
...
unsigned int fb;
unsigned int initializedTexture2D_1;
unsigned int initializedTexture2D_2;

GLenum buffs[] = {GL_COLOR_ATTACHMENT0,
GL_COLOR_ATTACHMENT1};

glGenFrameBuffer(1, &fb);
glBindFramebuffer(GL_FRAMEBUFFER, fb);

glFramebufferTexture2D(GL_FRAMEBUFFER,
GL_COLOR_ATTACHMENT0, GL_TEXTURE2D, initializedTexture2D_1, 0);

glFramebufferTexture2D(GL_FRAMEBUFFER,
GL_COLOR_ATTACHMENT0, GL_TEXTURE2D, initializedTexture2D_2, 0);

glDrawBuffers(2, buffs);

// render calls
...
```

```
#version 300 es

layout(location = 0) out lowp vec4 color;
layout(location = 1) out highp vec4 normal;

in lowp vec4 v_color;
in highp vec4 v_normal;

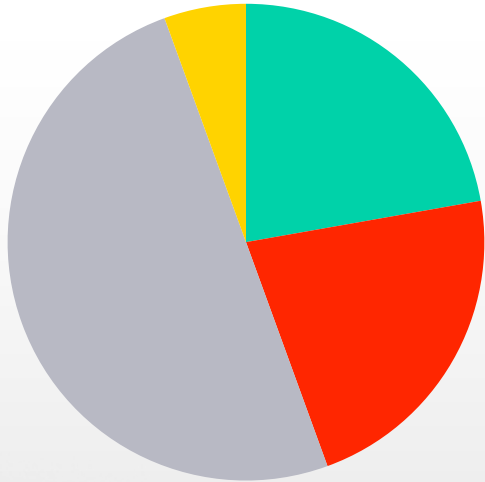
main() {

    color = v_color;
    normal = v_normal;

}
```


Cost-Benefit ratio

Cost



■ Instanced Rendering ■ Multiple Render Targets
■ Boolean Occlusion Queries ■ ETC2/EAC

Benefit



Challenges

- Implementation in existing engines is not trivial
- Changes to the production pipeline are needed
- In some situations OpenGL ES 3.0 features don't result in better performance
- MRTs needs to be understood by the graphical department as well
- OpenGL ES 3.0 devices are currently sparse → Support for both ES2/ES3

Opportunities



- **Better performance**
- **Smaller energy footprint**
- **OEMs love to see the latest innovations used by the developers**
- **Gap between current consoles and mobile devices getting smaller**
- **Through extensions some 3.0 features are available on current generation hardware**
 - `GL_EXT_occlusion_query_boolean`
 - `GL_NV_draw_instanced / GL_NV_instanced_arrays`

Cutting edge next-gen graphics

Useful links

http://www.khronos.org/opengles/3_X/

<http://www.imgtec.com/powervr/insider/sdkdownloads/index.asp>

<http://malideveloper.arm.com/develop-for-mali/features/opengl-es-3-0-developer-resources/>

<https://developer.qualcomm.com/mobile-development/mobile-technologies/gaming-graphics-optimization-adreno/tools-and-resources>

<https://developer.nvidia.com/tegra-resources>

<http://developer.android.com/tools/sdk/ndk/index.html>

Thank you for listening!

m.hehmeyer@fishlabs.net

FOLLOW US ON:

<http://www.twitter.com/fishlabs>

<http://www.facebook.com/fishlabs>

JOBS:

jobs@fishlabs.net

